AD-A055 092 MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTE--ETC F/G 5/10
LEARNING BY CREATING AND JUSTIFYING TRANSFER FRAMES,(U)
APR 77 P H WINSTON N00014-75-C-0643

UNCLASSIFIED AI-M-414A NL

| OF |

AD
A055092

END
DATE
FILMED
7-78
DDC

FOR FURTHER TRAN

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

12

14

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AI-M-414A ~~AIM 414A~~	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) 6 LEARNING BY CREATING ~~&~~ and JUSTIFYING TRANSFER FRAMES		5. TYPE OF REPORT & PERIOD COVERED MEMO
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) 10 Patrick H./Winston		8. CONTRACT OR GRANT NUMBER(s) 15 N00014-75-C-0643
9. PERFORMING ORGANIZATION NAME AND ADDRESS Artificial Intelligence Laboratory 545 Technology Square Cambridge, Massachusetts 02139		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Advanced Research Projects Agency 1400 Wilson Blvd Arlington, Virginia 22209		12. REPORT DATE 11 Apr[redacted]977
		13. NUMBER OF PAGES 12 38 P.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research Information Systems Arlington, Virginia 22217		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited.

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Artificial Intelligence
Cognitive Science
Computer Learning
Frame System
Information Processing Psychology

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Learning is defined to be the computation done by a student when there is a transfer of information to him from a teacher. In the particualr kind of learning discussed, the teacher names a <u>source</u> and a <u>destination</u>. In the sentence, "Robbie is like a fox," <u>fox</u> is the source and <u>Robbie</u> is the destination. The student, on analyzing the teacher's instruction, computes a kind of filter called a <u>transfer frame</u>. It stands between the source and the destination and dtermines what information is allowed to pass from one

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

447483

KC

BLOCK 20 CONCLUDED

to the other.

Computing the transfer frame requires two steps: hypothesis and evaluation. In the hypothesis step, potentially useful transfer frames are produced through an analysis of the information in the source and its immediate relatives. For Robbie, a robot, the way it compares with other robots would be noted. In the evaluation step, the better of the hypothesized frames are selected through a study of the destination frame, its relatives, and the general context.

Some source-destination pairs may be generated by the student acting alone. There is also the possibility of making notes that are useful in deciding if conclusion makes sense.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

AIM-414A

April 1977
Revised January 1977

LEARNING BY CREATING AND JUSTIFYING TRANSFER FRAMES

by

Patrick H. Winston

ABSTRACT

In the particular kind of learning discussed in this paper, the teacher names a *destination* and a *source*. In the sentence, "Robbie is like a fox," *Robbie* is the destination and *fox* is the source. The student, on analyzing the teacher's instruction, computes a filter called a *transfer frame*. The transfer frame stands between the source and the destination and determines what information is allowed to pass from one to the other.

Creating the transfer frame requires two steps: *hypothesis* and *filtering*. In the hypothesis step, potentially useful transfer frames are produced through an analysis of the information in the source and its immediate relatives. For a fox, the transfer frames are created through an analysis of the way foxes compare with other small mammels. In the filtering step, the better of the hypothesized frames are selected through a study of the destination frame and its relatives. For Robbie, the robot, the filtering is done by comparing Robbie with other robots.

Each time something is learned, the student takes notes on what happened. These notes are used later to justify subsequent conclusions. Also, the notes sometimes make it possible for the student to create his own source-destination pairs by himself.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-75-C-0643.

THE PROBLEM

Normally both the student and the teacher must do some work when there is a transfer of knowledge between them. The amount of work done by the two participants in the transfer can vary between two extremes, however. As illustrated in figure 1, there is a spectrum starting with learning by being programmed, moving through learning by being told, extending through learning by studying samples, and ending with learning by self-sufficient discovery.

This paper concentrates on learning by studying teacher-supplied samples. It offers a theory of learning by hypothesizing and filtering certain structures that will be called transfer frames. There is a sense that real learning is taking place because the student participates vigorously in the knowledge transfer process, working hard to establish just what the teacher is trying to convey.

The Approach Stresses Attention To Competences Rather Than To Particular Algorithms

Since learning is such a broad, complex phenomenon, it is sensible to start by being precise about the nature of the attack. This is an adaptation of the approach used by Marr in his work on vision [Marr]:

- First, it is necessary to observe or define some learning competence to be understood.
- Second, a representation is selected or invented that is capable of capturing the knowledge to be learned.
- Third, the first and second items are translated into a precisely defined computation problem to be solved.
- Fourth, algorithms are devised that perform the desired computation.
- And fifth, the results so far are validated by successful computer implementation and experimentation.

All this seems obvious, but there are strong temptations that often throw research out of proper perspective. One such temptation results in being caught up with an attraction to a particular representation. Worse yet, there may be an attachment to some particular algorithm, with a corollary failure to understand that many algorithms usually can be

devised once a computation problem is properly laid out.

Therefore, let us begin by concentrating on the definition of a kind of learning competence. Then we will turn to the the selection of a representation that seems appropriate and to the details of the algorithms which have been devised, implemented on a computer, and used in experiments.

Understanding Learning Requires Restricting What The Teacher Can Do

In previous work on learning, I explored how a computer can learn from a series of samples of simple blocks world structures like those in figure 2 [Winston]. The steps involved in using each sample are these:

- First, the computer analyzes the sample, producing a description in terms of objects, their properties, and the relations linking them. Normally samples that are not examples are the most valuable. These are called near misses.

- Second, the computer compares the description of the sample against the description of the concept as known so far. The comparison produces a list of differences.

- Third, the differences are ranked. If the teacher has shown a near miss, rather than an example, then the highest ranking difference is hypothesized to be the reason that the near miss is a loser.

- Fourth, changes are made to the concept description in response to the differences observed. This means that ordinary relationships are changed to MUST- and MUST-NOT- forms.

Thus, in the example shown in the figure, the computer learned that an arch consists of three bricks, one of which must be lying on top of two others that must not touch each other. Identification is done by comparing descriptions of unknowns with the finished concept descriptions. Identification of an unknown with a particular concept fails when MUST- type relationships are missing or MUST-NOT- relationships are present.

Of course it would be easy to write a program capable of absorbing facts about blocks world structures by being told directly, without requiring the use of samples. Such a program would not be as interesting, however, if the point is to probe into the more advanced competence that can be used by a learner. This deserves stress:

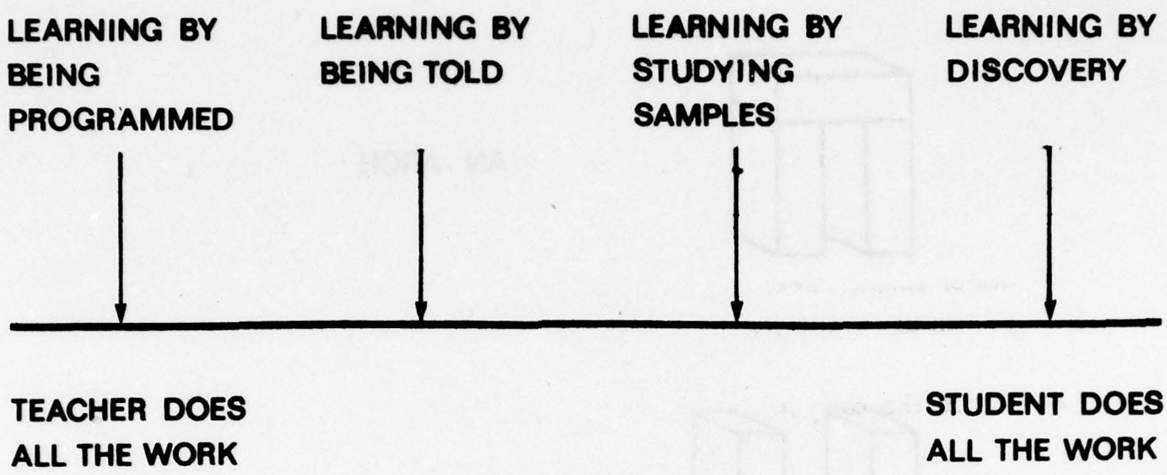

Figure 1. Learning may involve little work on the part of the learner or a lot. For there to be genuine learning, most people demand that the learner actively participate in the learning process. The simplest learning, really not learning at all, is learning by being programmed, with the learner doing nothing save submitting to the program surgery performed by the teacher. Learner participation is necessary when the learning is by being told or by understanding a series of samples. In the extreme, the participation of the learner is total, to the exclusion of the teacher, and there is learning by independant discovery.

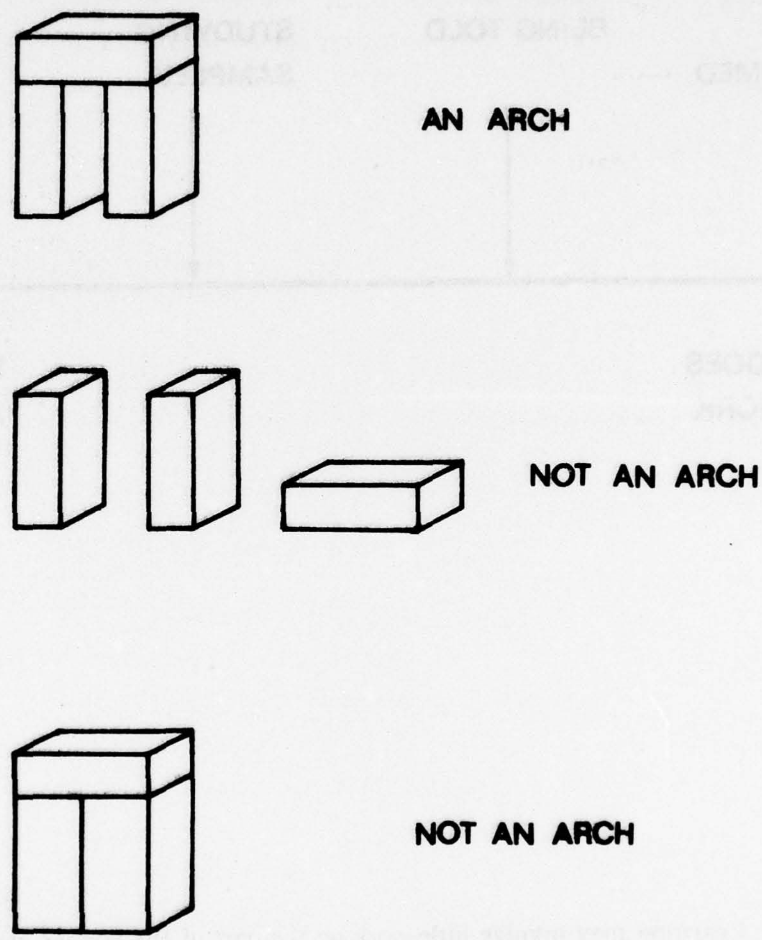

Figure 2. Learning in the blocks world using near misses. Each near-miss sample in this sequence identifies a particular part of the general arch description as essential.

- Learning requires a range of competences. To probe into some particular competence, it is necessary to turn off others. This can place seemingly unnatural restrictions on what the teacher can do.

In this paper, in particular, one may occasionally wonder at the almost cryptic nature of the teacher-student interaction forced by a need to isolate a single part of our learning competence.

The Focus Is On Learning Through Simile-Like Samples

Consider the following statement:

Robbie is a robot.

Hearing this, a human or computer student assumes some facts about Robbie and becomes curious about others. Robbie is probably made of metal, and it would be interesting to know if he is intelligent. Now consider these:

Robbie has a very high degree of cleverness.

Robbie is clever.

Robbie is clever like a fox.

Robbie is like a fox.

This paper is concerned with how we make sense of "Robbie is like a fox." At first this may seem strange since the other phrasings are certainly more precise. Many people, when hearing "Robbie is like a fox," would want to ask "In what way?" But forbidding such questions and limiting the teacher to "Robbie is like a fox," is necessary in order to disable the direct assimilation mechanisms that otherwise would mask something else:

- In the previous work on learning about arches and other simple blocks world structures, comparison of two descriptions was a key step. In this paper, comparisons are also important, but the comparisons are not between the things that the teacher names explicitly. Instead, the focus is moved to the close relatives of those teacher-named things.

The claim is that when we hear "Robbie is like a fox," we make use of what we know

about other robots and other common small animals of the woods to learn that Robbie is clever.

We are now in a position to present an introductory overview of the competence to be understood, the representation used, the computation problem, the resulting algorithm, and the validation process:

The competence. The central competence to be understood is the competence to absorb simile-like instruction. A secondary competence of interest has to do with curiosity. Given that Robbie is clever, we may wonder, for example, if Robbie is also like Suzie, another robot already known to be clever.

The representation. Of the many representations available now, the frames representation seems best suited in terms of the point of view that it encourages [Minsky]. Roughly, a frame is a chunk of knowlege describing something in terms of its properties. Here, for example, is a frame describing a fox:

FRAME NAME	SLOT	VALUE
FOX	A-KIND-OF	SMALL-MAMMAL
	COLOR	RED
	CLEVERNESS	VERY-HIGH

The frame name identifies what is to be described. Each of the properties that constitute the description is conveyed by a so-called slot-value combination.

Strictly speaking, the frame idea is a generalization of the much older property list idea, and it would nearly suffice in this paper to talk about atoms and properties, rather than frames and slots. The newer terminology is used for two reasons: first, some of the points of generalization will be incidentally introduced and used; and second, speaking of frames seems to imply an important commitment to knowledge chunking that is not implied when speaking of atoms and property lists.

Of course one objection to thinking in terms of frames is that the resulting programs can learn nothing that is not expressible in terms of frames. This seems true, but not particularly confining. The world of objects, classes, and processes that can be described in terms of frames seems amply large for useful learning research.

The computation problem. The key computation problem, therefore, is to fill frame slots using information given by a teacher in the form of simile-like instructions.

An algorithm. Here is the essence of an algorithm, to be described in detail later, that accomplishes the computation required to deal with simile-like instruction:

- The teacher's simile determines a *destination frame* and a *source frame*. In the sentence, "Robbie is like a fox," *Robbie* is the destination and *fox* is the source. The student, on analyzing the teacher's instruction, computes a filter called a *transfer frame*. It then stands between the source and the destination as in figure 3, determining exactly what slot-value combinations are allowed to pass from one to the other.

- Computing the transfer frame requires two steps: *hypothesis* and *filtering*. In the hypothesis step, potentially useful transfer frames are produced through an analysis of the information in the source frame and its immediate relatives. For a fox, other small common forest mammals would be used. In the filtering step, the better of the hypothesized frames are selected through a study of the destination frame and its relatives, together with the things learned in previous instruction. For Robbie, a robot, the way it compares with other robots would be noted.

This preview is given only to provide a flavor. Much more will be said about these procedures as well as others that deal with justification of transfers and internal generation of transfer possibilities.

Validation. The procedures described in this paper have been implemented and tested on the examples to be given. Exceptions are clearly noted. When the words *teacher* and *student* are used, the following is to be understood: the *teacher* is a human instructor, and the *student* is an experimental collection of algorithms implemented as computer programs. The programs are in LISP. No claims are made about psychological validation.

HYPHOTHESIS AND FILTERING

In a moment, we will look inside the boxes in the flowchart given in figure 4. In so doing we will uncover the details of an algorithm that performs some simple learning that is in accord with the proposed points of competence. To keep our own knowledge from getting too much in the way of thinking about the ideas, a semantically deprived world is used for most of the explanation. A consequence is that we, too, will have to work at understanding what is to be learned. Occasionally, our ideas may disagree with those of the program since both we and the program are working with limited information and we both therefore form somewhat shaky conclusions.

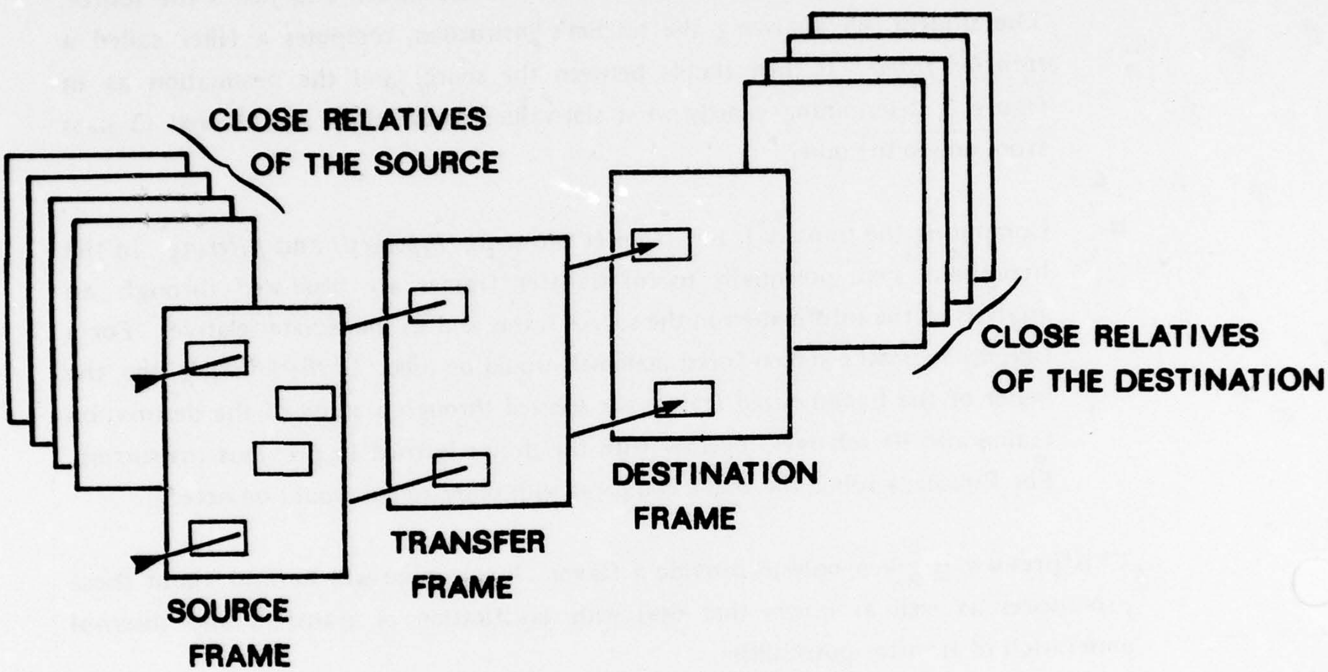

Figure 3. The basic idea behind the theory of learning presented in this paper. The teacher specifies a source and a destination and possibly the slots that are relevant. The student analyzes the source, the destination, and other aspects of the situation to discover and use a transfer frame.

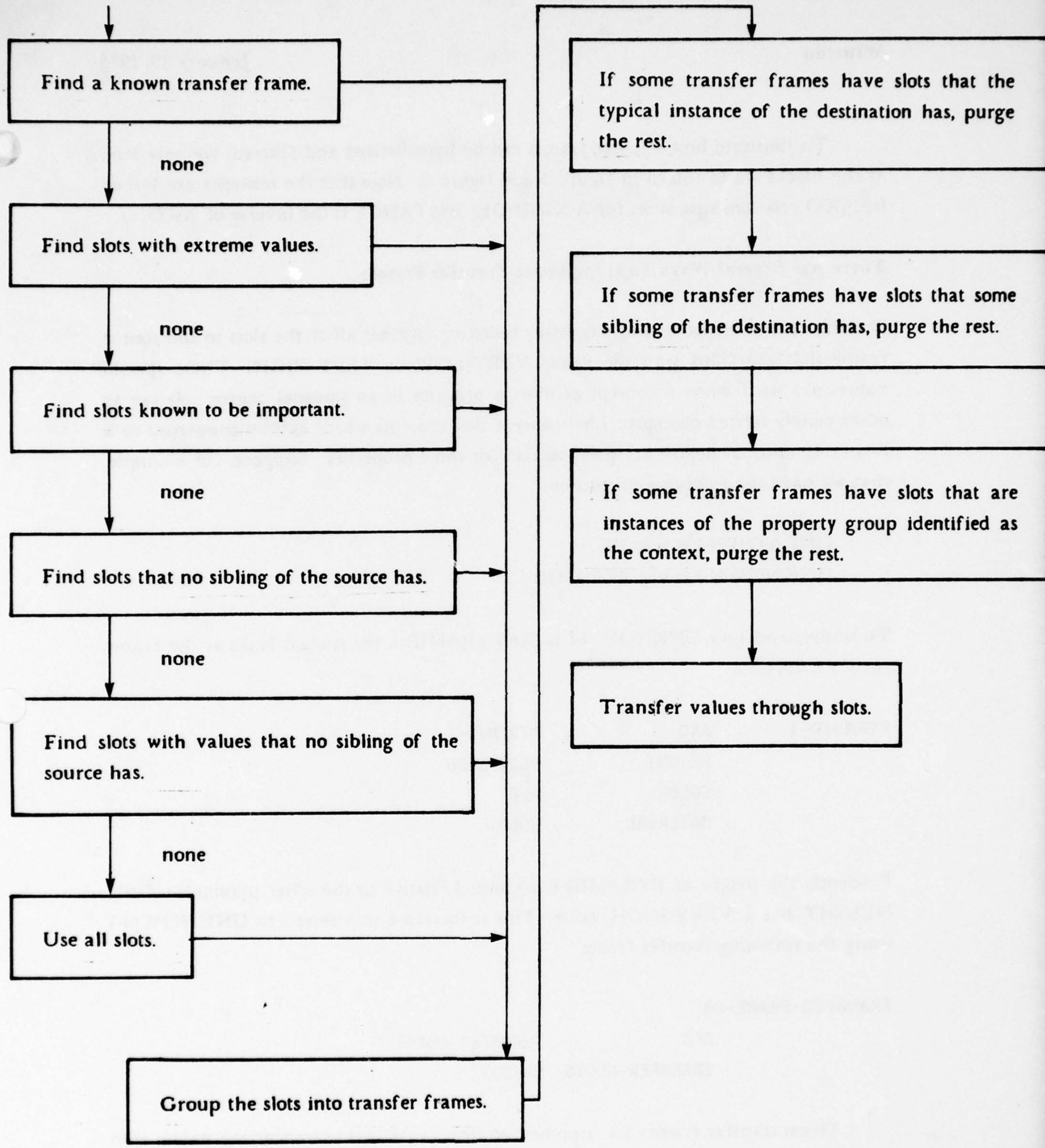

Figure 4. Overall organization of the hypothesizing and filtering methods. Hypothesizing methods are tried until one produces one or more slots that are not filled in the destination. After grouping into transfer frames, all filtering methods are used in an effort to reduce the number of surviving transfer frames. Filters have an effect only if they recommend dropping some, but not all of the transfer frames that they see.

To illustrate how transfer frames can be hypothesized and filtered, we now look at the blocks world shown in figure 5 and figure 6. Note that the concepts are linked by AKO relationships, short for A-KIND-OF. INSTANCE is the inverse of AKO.

There Are Several Ways To Hypothesize Transfer Frames

Transfer frame hypothesizing begins by collecting together all of the slots in the source frame that are filled with the values VERY-LOW or VERY-HIGH. These special values are used when a concept exhibits a property to an unusual degree relative to other closely related concepts. The theory is that concepts which exhibit properties to a relatively unusual degree are good sources for those properties. Suppose, for example, that we have the following instruction:

UNKNOWN-1 is a BOX.
UNKNOWN-1 is like PYRAMID-1.

To understand how UNKNOWN-1 is like PYRAMID-1, the student looks at the frame for PYRAMID-1:

```
PYRAMID-1        AKO              PYRAMID
                 HEIGHT           VERY-HIGH
                 COLOR            RED
                 MATERIAL         WOOD
```

Evidently the height of PYRAMID-1 is unusual relative to the other pyramids. Only HEIGHT has a VERY-HIGH value. This is therefore transferred to UNKNOWN-1 using the following transfer frame:

```
TRANSFER-FRAME-88
                 AKO              TRANSFER-FRAME
                 TRANSFER-SLOTS   HEIGHT
```

Often transfer frames are hypothesized that would lead to tranferring values into slots that already have values. There are two ways to handle such situations. First, the student may plan to add the new values to the old ones, perhaps after checking to be sure that the slots involved can take multiple values and that the new values do not conflict with old ones. Second, the student may reject the proposed transfer hypotheses immediately without any checking. In so doing, the student assumes that the teacher knows which slots have values and that the teacher never wants to add a value to a slot

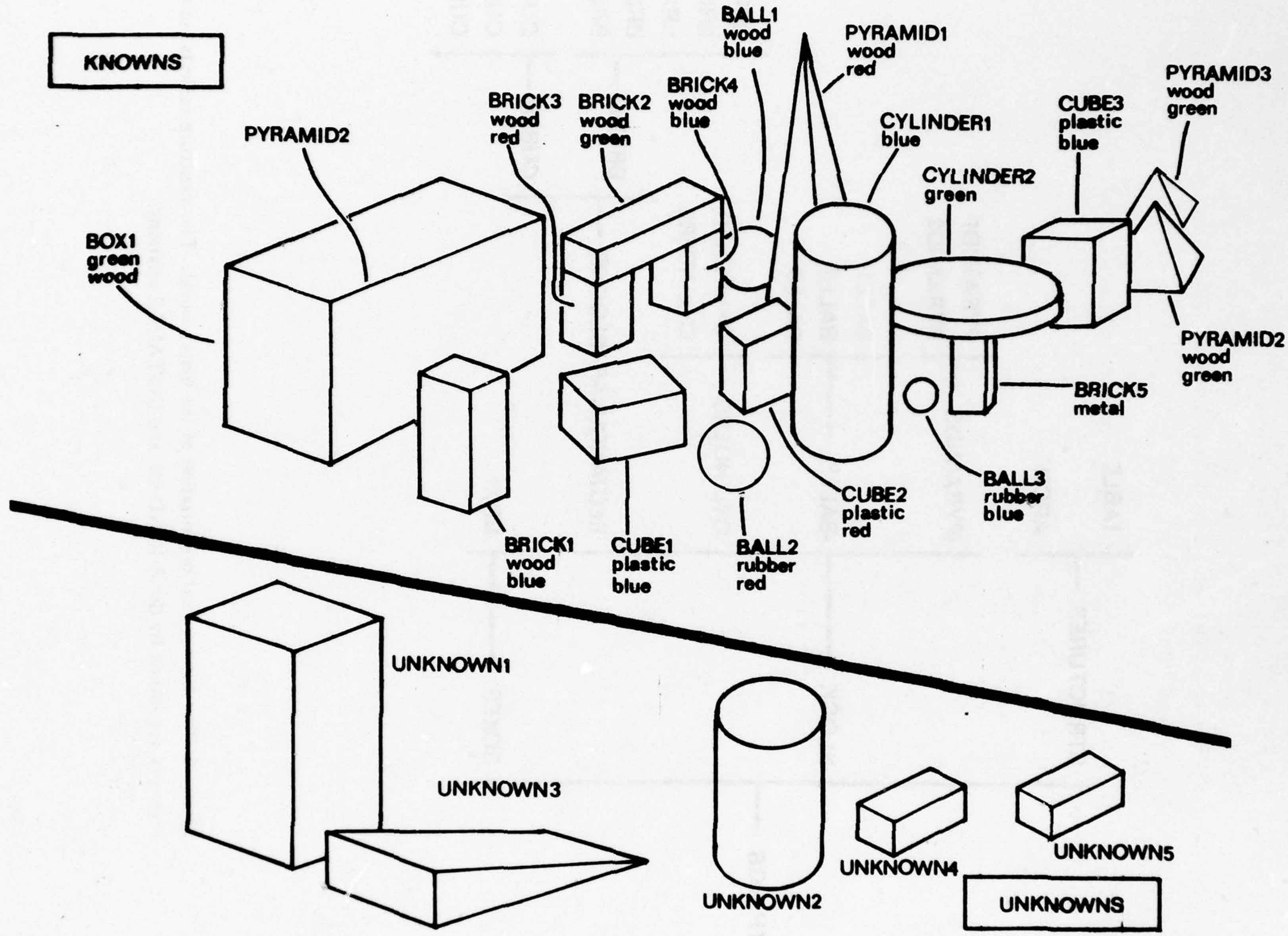

Figure 5. The objects in the blocks world.

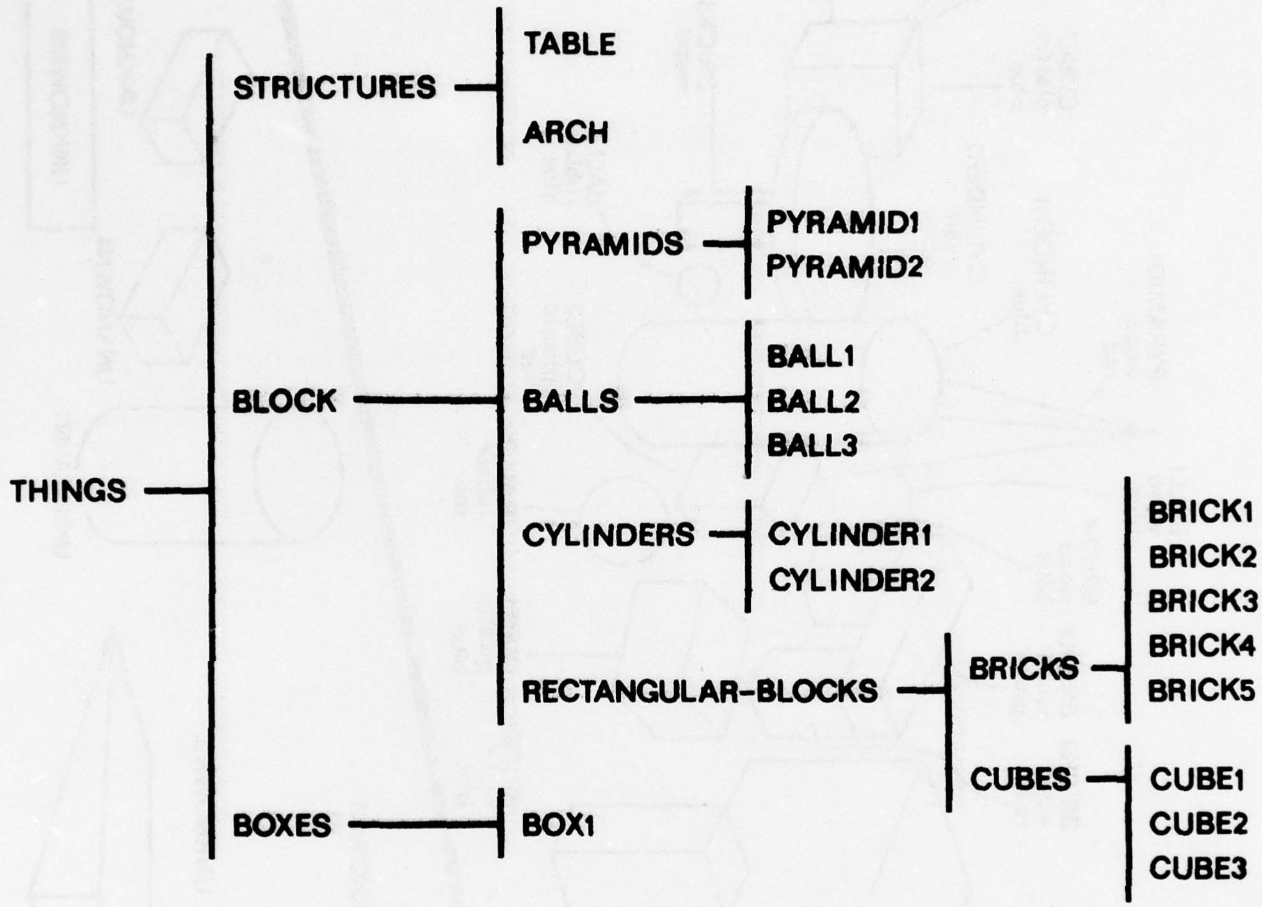

Figure 6. The hierarchical organization of the blocks world. The structure reflects how concepts are linked by the A-KIND-OF and INSTANCE relations.

that already has one. This is reasonable if the slots involved can only take one value and if there is some way that the teacher can know something of what the student already knows, perhaps by way of remembering what has been taught recently. In the implementation, the student rejects without checking, a choice selected strictly for implementation ease. No doubt it would be more natural for the student to do some checking and to add a value if possible, perhaps making an appropriate remark to the teacher about the result.

In any event, when the first method fails to find a viable transfer frame, others are tried until one works.

The second method again searches for important slots, but this time on the basis of global knowledge. Slots whose own descriptive frames contain VERY-HIGH in their IMPORTANCE slots are deemed globally important, and they are all collected. The slot PURPOSE, for example, is globally important. Consequently the following results in learning that UNKNOWN-1 is for storage.

UNKNOWN-1 is like BOX-1.

Inspection of the BOX-1 and PURPOSE frames shows why:

BOX-1	AKO	BOX
	COLOR	GREEN
	MATERIAL	WOOD
	PURPOSE	STORAGE
PURPOSE	AKO	FUNCTIONAL-PROPERTY
	IMPORTANCE	VERY-HIGH

Having dispensed with slots filled with exceptional values and slots known to be globally important, the next method concentrates on slots that are filled in an unusual way for concepts in the same class as the source. Consider the following descriptions of the balls:

BALL-1	AKO	BALL
	SIZE	MEDIUM
	COLOR	BLUE
	MATERIAL	WOOD

BALL-2	AKO	BALL
	SIZE	MEDIUM
	COLOR	RED
	MATERIAL	RUBBER
BALL-3	AKO	BALL
	SIZE	MEDIUM
	COLOR	BLUE
	MATERIAL	RUBBER

In BALL-1, the MATERIAL slot would be judged important because BALL-1 is one of three balls, BALL-1, BALL-2, and BALL-3, and of these, only BALL-1 has WOOD in the MATERIAL slot, which for balls is unusual. For BALL-2, the COLOR slot would be judged important because BALL-2 alone has a value in the COLOR slot that differs from the others. Consequently either of the following is like saying that UNKNOWN-1 has WOOD in the MATERIAL slot:

UNKNOWN-1 is like BALL-1 rather than BALL-2 or BALL-3.
UNKNOWN-1 is like BALL-1.

In the first sentence, the teacher supplies the relatives against which BALL-1 must be compared. In the second, the student must find them, but finding them is a simple matter of getting BALL-1's siblings from its parent's INSTANCE slot.

Now suppose that we move to UNKNOWN-2 and offer the following information:

UNKNOWN-2 is a CYLINDER.
UNKNOWN-2 is like BRICK-1.

BRICK-1, unfortunately, is rather undistinguished:

BRICK-1	AKO	BRICK
	SIZE	MEDIUM
	COLOR	BLUE
	MATERIAL	WOOD

Consequently, none of the hypothesizing methods given so far find anything, and the learner must simply gather up all the slots, hoping there will be some way of bringing more knowledge to bear later.

Note that after all of the slots are collected, they could be assembled together into a single transfer frame or into a set of transfer frames, one for each slot. Neither of these possibilities seems best because it seems better to group the slots together according to the property categories involved. The argument for grouping is that similarity with respect to one property weakly implies similarity with respect to other closely related properties. Grouping is done in the implementation.

In the current example, grouping does nothing since BRICK-1's slots, SIZE, COLOR, and MATERIAL, belong in distinct property groups as figure 7 shows. They therefore form three distinct transfer frames.

There Are Several Ways To Filter The Transfer Frames

When more than one transfer frame is hypothesized, it is up to the filtering methods to narrow the field. Several of these methods examine relatives of the destination, looking carefully for evidence that can pull the better transfer frames out of the pack. Consequently, in the current example, it is important to know that UNKNOWN-2 is a kind of cylinder and that CYLINDER-1 and CYLINDER-2 are too:

UNKNOWN-2	AKO	CYLINDER
CYLINDER-1	AKO	CYLINDER
	COLOR	BLUE
	SIZE	HIGH
CYLINDER-2	AKO	CYLINDER
	COLOR	GREEN

Evidently cylinders typically have a color but do not have one particular color. Said another way, there is typically a color slot, but there is no particular value typically resident in that slot. The *typical instance* is a frame created to record such facts after they are derived through a statistical look at the instances. The first of the next two frames indicates that TYPICAL-CYLINDER describes the typical cylinder attached by the INSTANCE relation to CYLINDER. The second specifies that only the COLOR slot and not its contents are typical.

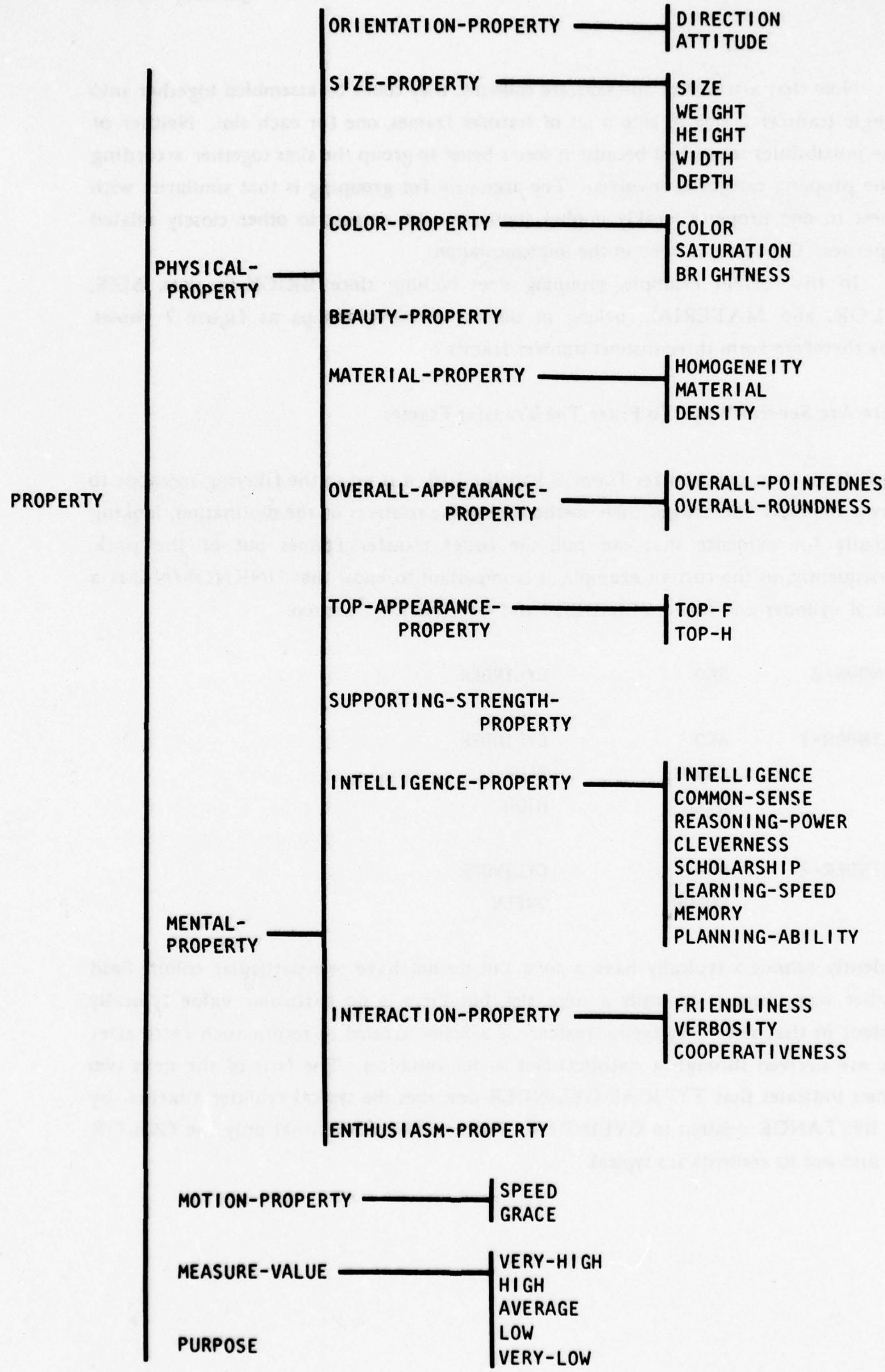

Figure 7. The hierarchical organization of the properties used in the examples.

```
CYLINDER            AKO                 THING
                    INSTANCE            CYLINDER-1
                                        CYLINDER-2
                                        UNKNOWN-2
                    TYPICAL-INSTANCE
                                        TYPICAL-CYLINDER

TYPICAL-CYLINDER
                    COLOR
```

Note that the appearance of a slot or slot-value combination in a typical instance frame means something quite different from what the same combination means when it appears in an ordinary frame. In the TYPICAL-CYLINDER frame, the slots and slot values record statistics on the immediate descendants of the associated node. In the CYLINDER frame, the slots and slot values indicate inheritable facts that are generally correct for all descendants from the node where they are found. (Certainly it might be reasonable to move things from the typical instance frame to the frame whose immediate descendants it describes, but how and when to do such movements has not been studied.)

Typical instance information is computed as follows: first, if a slot-value combination appears in more than some fraction of the instances, that combination goes into the typical instance; and second, if a slot appears in more than some fraction of the instances, but is not filled uniformly enough to pass the first test, it goes into the typical instance without a value. At the moment, both thresholds are set at 65%. This leads directly to the conclusion that the typical thing in the cylinder class has some color. Returning to the example, the first transfer frame filtering method exploits this typical instance information to pick out the transfer frame with the COLOR slot since the typical instance indicates that color is a commonly filled slot, one that is therefore wanted, in some sense, by the destination.

This seems to be involved when we understand things like "Her hair is like the wheat in the field." We assume that her hair is blonde or dry, not that it is good to eat, because color and texture are properties that hair typical has, while nutritional value is not.

As of now, in any case, we have the following frames:

BRICK-1	AKO	BRICK		
	SIZE	MEDIUM		
	COLOR	BLUE		
	MATERIAL	WOOD		
UNKNOWN-2	AKO	CYLINDER		
	COLOR	BLUE	TRANSFERRED-FROM	BRICK-1

Note that the COLOR slot of UNKNOWN-2 has the BLUE value augmented by a comment specifying where the value came from. This transferred-from comment is always placed when somthing is learned. The ability to attach a comment to value is a feature of the frame language that happened to be used in the implementation [Roberts-Goldstein].

Now suppose the teacher repeats the following statement:

UNKNOWN-2 is like BRICK-1.

Only the slots SIZE and MATERIAL emerge because COLOR is already filled. These form two frames, neither of which is better than the other with respect to the typical instance. Consequently another, weaker, method is used. This other method notes that some sibling of UNKNOWN-2 has a SIZE slot, namely CYLINDER-1. On the other hand no sibling has a MATERIAL slot. Hence the evidence favors using SIZE since it is more likely to apply than MATERIAL. Evidently UNKNOWN-2 is medium in size.

Next, to expose still another filtering method, let us consider the following pair:

UNKNOWN-3 is a WEDGE.
UNKNOWN-3 is like CUBE-1.

Assume that nothing more is known about UNKNOWN-3 and that CUBE-1 is described as follows:

CUBE-1	AKO	CUBE
	SIZE	MEDIUM
	COLOR	BLUE
	MATERIAL	PLASTIC

Just three frames are created, one each for SIZE, COLOR, and MATERIAL as in the example using BRICK-1. Now, however, there are no known relatives of UNKNOWN-3, so none of the previous filtering methods work. The decision, given

that the sequence is connected, goes to the frame that is most in keeping with the context determined by the last transfer. The last transfer involved size, so this one will too. (Actually the context is always reset to be the node in the property tree just above the last slot used. Consequently the context established is SIZE-PROPERTY, as shown in figure 7, and anything from the group SIZE, WEIGHT, HEIGHT, WIDTH, or DEPTH passes.)

This concludes the discussion of filters for the moment. A little more will be said later. Certainly the implementation is preliminary, and many changes may be found appropriate.

Near Misses May Generate Transfers As Well As Examples

Previous work on learning about arches stressed the idea of near misses, the samples that are not like the thing being described in some important way. The programs being described now do not deal with near misses only because the thrust is in the direction of dealing with new ideas, not because the old ones have been superseded. Indeed it is fairly clear how near-miss action could be incorporated into the current system:

- Use the same hypothesis methods without change.
- Use the same filtering methods, except that slots are not to be rejected merely because they happen to be filled in the destination frame.
- Revise the way the transfer frame is used to carry slot information from the source to the destination.

Unlike the properties in a property list, the slots in a frame can have more than just a value associated with each slot. The value is just one of many possible facets. This feature is useful in handling near misses because instead of adding to the VALUE facet, it is possible to add to a MUST-BE or MUST-NOT-BE facet. With this it would be possible to give the following to the student:

> An ARCH is *not* like a TABLE.

The expected result would be the placement of EAT and WRITE in the MUST-NOT-BE facet of the PURPOSE slot of ARCH. This would happen even if ARCH already had something in the VALUE facet of the PURPOSE slot.

Summary

Hypothesis methods concentrate on looking at the source and its context. The hypothesis methods are as follows:

- Use a remembered transfer frame. This method has not yet been described, but examples will be given later.
- Make a transfer frame using slots with extreme values.
- Make one using slots that are known to be important in general.
- Look for slots that are unique with respect to the source's siblings. This has not been described, but it is much like the next one, which has.
- Look for slots that have unique values in them with respect to the siblings.
- Use all of the source's slots.

All hypothesis methods weed out slots that are already filled in the destination, and all group the slots they find using the property heirarchy. The filtering methods focus on the destination and its context, together with the learning sequence. They are as follows:

- Prefer transfer frames that have slots that are present in the typical instance associated with the destination.
- Prefer those that have slots that some sibling of the destination exhibits.
- Prefer those that are in the same property group that was involved in the last transfer.

JUSTIFICATION AND CURIOSITY

Once a transfer frame is in hand, then it is necessary to decide if using it really makes sense given all that is known about the slots it will effect. If it is used, then there is the further question of whether the new knowledge gained about the destination should trigger the student into further, internally generated speculation.

There Are Several Ways To Justify The Transfer Frames

Hypothesizers and filters do their jobs by examining the siblings of the source and the destination and the slot last involved in a transfer. Once they do their job, there are further ways to examine the surviving transfer frames to test their suitability. These are called justification methods. They are miscellaneous methods that are somewhat tentatively held separate from the filters for one small reason: the justification methods are used in an absolute, rather than a relative way. The justification methods are applied to transfer frames individually and without regard to what other transfer frames may be available. The filters do nothing if they find that all the hypothesized transfer frames do equally poorly. They are applied to reduce the size of the group of transfer frames that come out of the hypothesis activity.

First, of course, the student can justify a transfer frame by asking the teacher directly if the frame is appropriate.

Second, the student can check for restrictions that would prevent the insertion of obviously wrong values. (Restrictions on slot values are remembered by placing predicates in the RESTRICTION facet of the slot.)

The third method is one that requires the student to take notes on why transfers seem to work and to create justification frames that can be matched against a proposed destination to see if the destination exhibits apparently essential slot values.

Suppose, for example, that the student knows CUBE-1 has MEDIUM in the SIZE slot and VERY-HIGH in the TOP-FLATNESS and TOP-LEVELNESS slots. Further suppose that the teacher gives this:

CUBE-1 is like TABLE.

The intent must be that it is possible to eat from or write on CUBE-1, just as it is with a table, since there are now the following frames:

```
TABLE           AKO               STRUCTURE
                PURPOSE           EAT
                                  WRITE
                HAS-PART          BRICK-5
                                  CYLINDER-2
                SIZE              MEDIUM
                TOP-FLATNESS      VERY-HIGH
                TOP-LEVELNESS     VERY-HIGH

CUBE-1          AKO               CUBE
                COLOR             BLUE
                MATERIAL          PLASTIC
                SIZE              MEDIUM
                TOP-FLATNESS      VERY-HIGH
                TOP-LEVELNESS     VERY-HIGH
```

PURPOSE is selected for transfer since it is known to be an important slot. After the transfer, the student, on the request of the teacher, looks to see how the source and the destination resemble one another. Then the student constructs a justification frame that reflects the similarity and remembers both the justification frame and the transfer frame:

```
TABLE           AKO               STRUCTURE
                .
                .                 TRANSFER-FRAME  TRANSFER-FRAME-78
                .

TRANSFER-FRAME-78
                AKO               TRANSFER-FRAME
                TRANSFER-SLOTS    PURPOSE
                TRANSFERRED-FROM
                                  TABLE
                TRANSFERRED-TO
                                  CUBE-1
                JUSTIFICATION-FRAME
                                  JUSTIFICATION-FRAME-31
```

JUSTIFICATION-FRAME-31		
	AKO	JUSTIFICATION-FRAME
	SIZE	MEDIUM
	TOP-FLATNESS	VERY-HIGH
	TOP-LEVELNESS	VERY-HIGH

TABLE has become a standard source of particular values for the PURPOSE slot, namely EAT and WRITE, through the skillful selection of circumstance by the teacher.

Now it is time to introduce a simple new hypothesizer that will always be the first to work. It will just look for values in the TRANSFER-FRAME slot. Whenever TABLE is a source, it will find a value, namely TRANSFER-FRAME-78.

Importantly, the student has a justification frame attached to this standard transfer frame. This justification frame must be a subframe of a proposed new destination if the new destination is to pass. In this example, for PURPOSE to be transferred to a destination from TABLE, the destination must have the SIZE, TOP-FLATNESS, and TOP-LEVELNESS slot values dictated by the justification frame, JUSTIFICATION-FRAME-31.

Now consider this:

CUBE-2 is like TABLE.

There will be a justified PURPOSE transfer if the student's CUBE-2 frame has the three key justification frame slots properly filled or if the student can get proper values from the teacher or by some other means.

In general, this is really only a mechanism for getting a first idea of why a given transfer is or is not justified. Further refinement of the justification frame is possible by direct telling or by fresh transfers to it as a destination from other justification frames.

While all this student note taking is going on, information is also added to the SIZE, TOP-FLATNESS, and TOP-LEVELNESS frames:

SIZE	AKO	SIZE-PROPERTY		
	TRIGGER-VALUE	MEDIUM	TRANSFER-SOURCE	TABLE
TOP-FLATNESS	AKO	TOP-APPEARANCE-PROPERTY		
	TRIGGER-VALUE	VERY-HIGH	TRANSFER-SOURCE	TABLE

TOP-LEVELNESS	AKO	TOP-APPEARANCE-PROPERTY			
	TRIGGER-VALUE	VERY-HIGH	TRANSFER-SOURCE	TABLE	

Of course the TRIGGER-VALUE slot for SIZE will become gorged far sooner than for TOP-FLATNESS and TOP-LEVELNESS since SIZE is a more common property. This means that SIZE will not be as useful as the other two with respect to the use of trigger values about to be described.

Filling A Slot May Induce Curiosity

It is reasonable for the student, having just learned something, to make conjectures based on the new knowledge. Often these conjectures will be wrong since they are generated internally using rather flimsy heuristic evidence. Hence it will be more important than usual to use the various justification methods to confirm the conjectures.

To see how one conjecture method works, suppose that UNKNOWN-4 has the following description:

UNKNOWN-4	AKO	BRICK
	SIZE	SMALL

The first conjecture method uses information in existing slot and justification frames. Suppose that the following is given next:

UNKNOWN-4 has VERY-HIGH in the TOP-FLATNESS slot.

From this, and with permission of the teacher to think a bit, it is reasonable for the student to examine the TOP-FLATNESS frame for clues about other properties of UNKNOWN-4. The TRIGGER-VALUE slot of TOP-FLATNESS contains the value VERY-HIGH along with a comment to the effect that the value was placed while constructing a justification frame involving a transfer from TABLE. Since VERY-HIGH in the TOP-FLATNESS slot evidently helped justify a transfer from TABLE in the past, it is reasonable for the student to try a transfer from TABLE again, this time to UNKNOWN-4. Thankfully the trigger value information exists only if a justification frame also exists. The student therefore has a justification frame that he uses to decide if the transfer makes sense, possibly asking the teacher some questions along the way about the slots that the justification frame specifies.

A second conjecture method uses siblings. Suppose that the following is given:

UNKNOWN-4 has BLUE in the COLOR slot.

Using this, the student may want to look for siblings of UNKNOWN-4 that are also blue with the hope that UNKNOWN-4 and such a sibling may be alike in other ways. Indeed, this happens in the implementation. Siblings with BLUE in the COLOR slot are collected, and the most typical one becomes a conjectured source.

The most typical blue sibling is determined by comparing all blue siblings with the typical instance using a frame similarity computation defined as follows:

<frame similarity between X and Y>
$= a/b$ for $b > 0$
$= 0$ for $b = 0$

where a is the number of slot-value combinations that appear in both X and Y and b is the number of slot-value combinations that appear in either.

If all of the slots in X and Y have different values, then the frame similarity will be zero. If all of the slots in X and Y have the same values, then it will be one.

In this example, both BRICK-1 and BRICK-4 are blue. BRICK-1 is judged the more typical of the two because the frame similarity between BRICK-1 and the typical instance frame for BRICK, TYPICAL-BRICK, is .66, whereas the frame similarity between BRICK-4 and TYPICAL-BRICK is only .5. The difference is the result of a PURPOSE-SUPPORT slot-value combination present in BRICK-4 but missing in BRICK-1 and TYPICAL-BRICK.

Thus, learning that UNKNOWN-4 is blue results in a transfer from BRICK-1 that would assert that UNKNOWN-4 is made of wood. Again, however, if any justification methods are available, they are used. Moreover, it would be more sensible to transfer only those properties from BRICK-1 that are closely related to the COLOR slot since the conjecturing method is so tenuous. This, however, has not been implemented in the existing system.

Making these conjectures is one kind of "curiosity." Another can come from obvious reaction to learning that an unknown is a kind of something. Consider this:

UNKNOWN-5 is a BRICK.

Without further fuss, it would make sense for the student to assume that UNKNOWN-5 has all the typical slot-value combinations that are in the typical instance frame, assuming that UNKNOWN-5 is of medium size and is made of wood as a result. But the student should also know that typical instances may have unfilled slots that get there when a slot is common but does not appear with the same value

often enough for a value to accompany it into the typical instance. At the moment, the teacher is asked to supply values for these slots either explicitly or by reference to a source with the proper value. For the example given, then, the following is printed:

I assume UNKNOWN-5 has the slot-value pairs
SIZE-MEDIUM and
MATERIAL-WOOD.

I am curious about a value for COLOR.

This seems related to the use of something like typical instances by Davis in his system that helps users write new production rules [Davis].

If the teacher supplies no direct answer to the students curiosity, it is nevertheless conceivable that the student may successfully fill the slot by guessing a suitable source. Two methods come to mind that parallel the methods just described for responding to a new slot instantiation. Unlike the other methods, neither have been implemented.

A first source to consider might be found by again appealing to justification-frame information. The justification-frame construction program already places JUSTIFIED-VALUE slot information just as it places TRIGGER-VALUE INFORMATION. For the table example, we have this:

```
PURPOSE          AKO              PROPERTY
                 JUSTIFIED-VALUE
                                  EAT    TRANSFERRED-FROM        TABLE
```

Wondering about the value for the PURPOSE slot of something, the student can use TABLE as a possible transfer source to be attempted. If there are many justified values in the slot's frame, then the student might well want to screen them by looking for the known purposes of the siblings of the destination frame. There is a greater chance that the destination will have a purpose similar to one of its siblings than to something entirely removed.

A second source to consider for transfer is the most typical sibling of the destination frame that has the slot in question filled. Generalizing, the student could move up the AKO tree, looking for a suitable sibling of the more remote ancestors, not just the parent, until one is found that a lot is known about. (Perhaps this is supported by the evidence that children like to use their knowledge about humans as a primary source for assumptions about other animals.)

The Blocks World May Be Deceptively Small

The small number of properties associated with each object may be a cause for some uneasiness. Is it possible that the examples work only because of the careful arrangement of the slots and their small number? Maybe. Indeed one important question to be addressed as the work goes on is that of how much complexity can be coped with before the system breaks down. In this connection, two points are probably worth observing:

- The fact that things are immersed in an AKO tree will tend to keep the clutter under control. Presumably most property values are obtained by defaulting to higher and higher level concepts.

- Good teaching normally requires using examples with relatively few prominent properties. Good examples are the ones for which the computation required for understanding is low.

Indeed the reason the simple physical world is a good source of general similes, some of which reach the social world, the mental world, and various expert problem solving worlds, may be because its simplicity makes the simile understanding problem easier.

EXAMPLES FROM THE ANIMAL WORLD

Animal world is shown in figure 8. We will use it first to review some basic hypothesis and justification ideas, then we will turn fleetingly to an example involving analogy, and finally, we will speculate on how similes might be generated.

Jack And Jill Can Be Described By Animal Similes

Consider this sample sequence:

Jack is like a fox.

Since fox has a very high value for cleverness, it is concluded that Jack does too. The context becomes intelligence and the use of the fox as a simile for cleverness will be noted.

Jill is like an elephant.

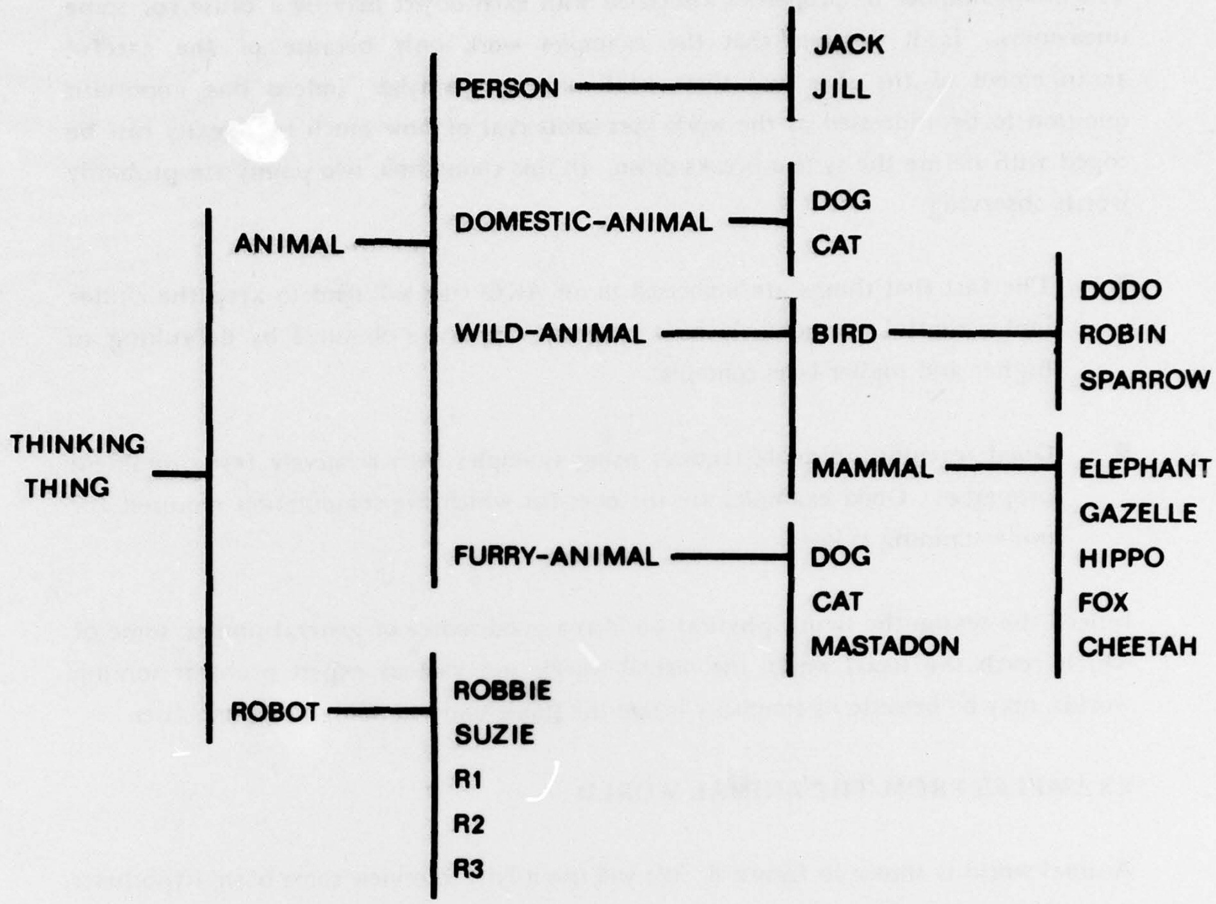

Figure 8. The hierarchical organization of a simple animal world.

Since an elephant has several slots, there are several possibilities, namely memory, weight, and grace. Good memory is the winner though, since the context is intelligence.

Jill is also like a cheetah.

Evidently Jill is fast. The context now has to do with motion properties.

Robbie is like an elephant.

The context now singles out grace and transfers a low value because the context now has to do with motion, not weight or memory.

Robbie is a robot.
Robbie is like an elephant.

Robbie already has a grace property. The transfer must have something to do with either weight or memory. Knowing that Robbie is a robot helps because the other robots have values in the memory slot but not in the weight slot. Evidently Robbie has a good memory.

Robbie is like an elephant.

The third time around, only weight is left. The context becomes size.

Now for the next example, suppose the frame for Suzie has the following information:

SUZIE	AKO	ROBOT
	INTELLIGENCE	MEDIUM
	MEMORY	HIGH
	COMMON-SENSE	MEDIUM
	REASONING-POWER	LOW
	VERBOSITY	LOW

These properties make two groups: one deals with intelligence, memory, common sense, and reasoning power, all aspects of the general notion of intelligence, and the other deals with verbosity, a dimension of personality. Consequently, the use of the Suzie frame in transfers may cause two transfer frames to be created, one for each of the two groups.

We are now able to understand still another filtering mechanism. Suppose that two transfer frames are indeed created when Suzie is a source and the following is given:

Sally has medium common sense.
Sally is like Suzie.

What properties of Suzie are preferred for the next transfer? Intelligence, memory, common sense, and reasoning power could be relevant or verbosity might be right. But since Sally's common sense is already known to be medium, the choice is to pass values through other slots that are in the same transfer frame with common sense, namely the intelligence, memory, and reasoning power slots.

Transferring intelligence, memory, and reasoning power information is the preferred action because having one fact about intelligence makes acquiring more a likely possibility. So far Sally has no personality properties, and it would be more risky to transfer through the verbosity slot.

A Transformation May Be Specified Directly Or By Analogy

Of course a value need not slither through a transfer frame unscathed. Generally, it may be subjected to some sort of value transformation. VERY-HIGH becomes VERY-LOW if MAKE-OPPOSITE is the transformation in effect. MEDIUM becomes HIGH if MAKE-MORE is the transformation. An APPLE becomes FRUIT by way of MAKE-GENERAL. Other, fancier things may be useful in making similes between worlds.

The name of the transformation may be directly specified, of course, as in the following fragment:

John is the opposite of a fox.

However, the transformation may be given by an analogy:

Jane resembles a fox in the same way John does.

After CLEVERNESS is found to be the slot involved in comparing JANE with FOX, it is a simple matter to test John against FOX, finding that MAKE-OPPOSITE is the implied transformation.

Testing the transfer frame using the analogy source and the analogy destination also can help filter out wrongly conjectured transfer frames that may have survived all

other filtering operations. It better be true that the same transformation applies to all of the slots in the transfer frame when it is used to compare the analogy source and analogy destination frames. Otherwise, chuck it out.

Note, incidentally, that the source, the destination, the analogy source, and the analogy destination may all be different. The teacher may or may not supply any of these four items, together with the transfer frame and the transformation, giving a total of 63 combinatorial possibilities, the bulk of which are probably absurd.

USEFUL PRINCIPLES

There has not been enough experiment with the programs and the ideas in them to know how much can be accomplished. Ideas have been illustrated, but more experiments and much larger, more completely specified domains are needed. Still, there is some hope that the following principles may hold:

The principle of representational parsimony. If knowledge of all kinds is represented uniformly, then all sorts of things will be subject to the same learning processes. Since objects, properties, and even things like justifications have the same representation, all can be learned about through transfer frames. With respect to domain, any in which the objects can be described in terms of frames is potentially a domain that learning using transfer frames can address.

The principle of expanding competence. The more that is known, the better learning should be, both in terms of speed and accuracy. Certainly speed and accuracy should increase with increasing knowledge when learning is by transfer frames since the more the student knows, the easier it is for the teacher to find lucid examples less subject to misinterpretation.

SPECULATIONS

Traces Could Be Used To Find Substitutes, Note Attributes, And Pass The Time

Look again at the example of the table transfer to the cube. Having noted that the transfer took place while both frames were observed to be of medium size and to have flat tops, a lot of information was recorded that might be used as follows:

- Having made the transfer to CUBE-1, idle time could be spent seeing if relatives of CUBE-1 are also like TABLE when compared through the JUSTIFICATION-FRAME, JUSTIFICATION-FRAME-31. If so, proceed to learn more by making the transfer through the transfer frame, TRANSFER-FRAME-78.

- To find something which would serve the same purpose as a table, note that the table's purpose was transferred earlier by using the table's recorded transfer frame and justification frame. See if anything in the physical vicinity satisfies the justification frame. Index into the frames in the vicinity, perhaps using the justification frame's slots.

- To find something whose purpose is to serve in writing, look into the frame for PURPOSE and note that TABLE has been a source of similes for writing. Get the TRANSFER-FRAME and JUSTIFICATION-FRAME information from TABLE.

Past Transfers Could Be Used In Generating Descriptions

The simplest ways to generate descriptions is to bounce back information previously digested. The system already leaves certain information behind to enable this.

First, when transfers are used to transfer information into a concept, the source is recorded. If Sam was said to be like a fox, it would be easy to say this:

Sam is very clever, like a fox.

Just having this would make conversation dull and full of triteness, but other devices could be used:

- When a frame has a slot filled with a VERY-HIGH or VERY-LOW value, the fact could be recorded in the slots frame by instantiation of the VERY-HIGH-VALUES or VERY-LOW-VALUES slots using the name of the frame where the exceptional value was recorded. This frame is then a possible transfer source.

- If a frame with an extreme value was used as a source before, it should be particularly good. It is better if it has not served as a transfer source contributing to the description of the thing to be described.

- If, in looking over what is to be said, there are many possible sources, the transfer generator can run its various source possibilities through the filters using its best guess about what the listener already knows about the concept being described. Clearly the best descriptions are the ones that allow rapid filtering down to the correct transfer frames. This means the sources specified will automatically tend to tell the listener facts the listener does not know and stick to a context, among other things.

- The transfer generator may decide it is folly to do the whole description as a chain of similes. Instead it may be better to specify explicitly a slot or a context from time to time.

- The transfer generator can bias itself by choosing sources from either pleasant categories (fields and flowers) or unpleasant ones (fire and brimstone).

- As an additional literary device, a pointer into the AKO tree should be maintained and transfer sources should be selected from the descendants of it. This would tend to help avoid inelegant mixing of similes.

SUMMARY

The path has been involved. Therefore it makes sense to put some of the key ingredients on display now, by way of summary:

The approach. I began by stressing the importance of understanding just what learning computations are to be understood and how they can be exposed by limiting the teacher to simile-like instruction. Then I offered a representation, posed a computation problem, gave some algorithms, and discussed an implementation.

Frame-like representation. A representation is a vocabulary of symbols and a set of conventions for arranging them to describe things. Obviously representation is a central issue in attempts to understand learning, for nothing can be learned unless there is a representation that can capture the new knowledge to be learned. Consequently, when a new and powerful representation is found, it is useful to examine it with a view toward addressing the competences exhibited by learners.

Hypothesizing and filtering transfer frames. The *destination* frame is the thing to be learned about. It has *slots* that may assume *values*. Values are supplied by a *source* frame that happens to have a slot value suited to the destination. Typically there are many possible ways the source may be like the destination. A *transfer frame* is a frame that stands between the source and the destination like a template and determines the information that is transferred. A key idea is that a student can generate these transfer frames dynamically by using a variety of common-sense methods that access what is already known. The good teacher, knowing how these methods work and having a rough model of what is already known by the student, can teach in a way that improves both the transfer rate and accuracy.

Using justification frames. Keeping track of specific properties that legitimize slot values in various ways is another way to judge a proposed transfer. The legitimizing slot-value combinations are stored in *justification frames*. These justification frames can be accumulated by experience. They can also be acquired and honed by dialogue with the teacher, just as other frames can be.

Grouping and the typical instance. Groups are important. Groups of things tend to have the same properties, not just a single, group-defining property. Consequently, an abstract *typical instance* can be constructed for a group. The typical instance consists of a number of slots and slot values that capture the essence of the group it describes. Typical instances are important because of their role in filtering transfer frames, in making assumptions about new objects, and in proposing sources autonomously. The notion of typical instance derives from earlier work [Winston] [Davis].

Similarly, groups of properties are important. The group dealing with size, for example, encompasses general size, weight, height, width, and depth. If one is mentioned, it helps to establish a context in which the others are expected.

REFERENCES

- Randall Davis, "Applications of Meta Level Knowledge to the Construction, Maintenance and Use of Large Knowledge Bases," PhD thesis, Stanford Artificial Intelligence Laboratory, AIM-283, 1976. Cited here for its interesting use of groups and typical members to help humans in the design of situation-action rules.

- David Marr, "Cooperative Computation of Stereo Disparity," *Science*,, Vol. 194, October 1976. Cited here for its description of Marr's approach to the computation problems of psychology.

- Marvin Minsky, "A Framework for Representing Knowledge," in *The Psychology of Computer Vision*, edited by Patrick Henry Winston, McGraw-Hill Book Company, New York, 1975. The paper in which the vocabulary of frames was largely established.

- Bruce Roberts and Ira Goldstein, "The FRL Manual," AIM-409, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1977. Describes the frame system used by the programs described here.

- Patrick Henry Winston, "Learning Structural Descriptions from Examples," in *The Psychology of Computer Vision*, edited by Patrick Henry Winston, McGraw-Hill Book Company, New York, 1975.